SenseCluster at SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection

Amaru Cuba Gyllensten
RISE
amaru.cuba.gyllensten@ri.se

Evangelia Gogoulou
RISE
evangelia.gogoulou@ri.se

Ariel Ekgren
RISE
ariel.ekgren@ri.se

Magnus Sahlgren
RISE
magnus.sahlgren@ri.se

Abstract

We (Team Skurt) propose a simple method to detect lexical semantic change by clustering contextualized embeddings produced by XLM-R, using K-Means++. The basic idea is that contextualized embeddings that encode the same sense are located in close proximity in the embedding space. Our approach is both simple and generic, but yet performs relatively well in both sub-tasks of SemEval-2020 Task 1. We hypothesize that the main shortcoming of our method lies in the simplicity of the clustering method used.

1 Introduction

The meaning of a word can vary not only with context, but also with time. The former phenomenon is commonly known as *context-sensitivity*, or, if the variation is of a more categorical nature, *polysemy*, whereas the latter phenomenon is captured by the term *diachronic semantic drift* (Kutuzov et al., 2018), or alternatively *lexical semantic change* (Schlechtweg et al., 2020). As an example, a term such as "beautiful" has one main meaning, but will nonetheless imply slightly different things depending on its context of use; "suit" on the other hand has several distinct meanings (e.g. as a verb or as a noun), while "mouse" has acquired a completely new meaning with the introduction of computer hardware. Of course, the distinction between context-sensitivity and polysemy is anything but clear-cut; this is a slippery theoretical slope, on which it is best to tread lightly. Even so, enabling the detection of such diachronic lexical semantic change across time could accelerate research in historical linguistics (Szymanski, 2017), and also initiate the development of decision-making systems that exploit diachronically shifting information (Rosin et al., 2017).

The backbone of a lexical semantic change detection system is word embeddings, which represent the meaning (or at least the *use*) of words. Different systems rely on various types of language models, nowadays predominantly distributional in nature. There is a comparably rich literature on distributional approaches to modeling diachronic semantic drift; examples include Sagi et al. (2008) Hamilton et al. (2016), and Yao et al. (2018). More complete overviews of existing diachronic semantic shift detection techniques is provided in Tahmasebi et al. (2018) and Kutuzov et al. (2018).

Contextualized language models constitute a recent breakthrough in the field of NLP (Devlin et al., 2018; Radford et al., 2018), by virtue of their ability to provide embeddings that are sensitive to a *specific* context of use, which is different from standard word embeddings that aggregate all of a word's contexts into one global representation. Another way of characterizing this difference is to say that contextualized language models provide token-based representations, while standard word embeddings are type-based. Motivated by the success of contextualized language models for handling polysemy and context-sensitivity, we investigate how such embeddings can be used to model diachronic semantic change; we use a contextualized language model as the basis of our proposed system for both sub-tasks of the Unsupervised Lexical Semantic Change Detection task, featured in SemEval 2020. More specifically, we produce contextualized embeddings for each occurrence of a term, and cluster these embeddings to arrive at a form of sense clusters. By leveraging multilingual contextualized representations, our approach

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 112–118
Barcelona, Spain (Online), December 12, 2020.

is agnostic to which language is used in the input corpus, and it does not rely on any specific information about when the corpus was written. Despite the simplicity of our approach, our system ranked 10^{th} in Subtask 1 and 8^{th} in Subtask 2.

1.1 Subtask description

1.1.1 Subtask 1

Subtask 1 is Binary Lexical Semantic Change, as defined in Schlechtweg et al. (2020). Given a set of terms and two corpora from two different time periods, the goal is to identify the terms with different sets of senses between the two corpora, and consequently time periods. The fact that the corpora are also provided in languages other than English pushes towards the direction of designing language-agnostic systems (Kutuzov et al., 2018). No labels are provided, therefore the system should be unsupervised. Annotations provided by Schlechtweg et al. (2020) will be used as ground truth labels in the evaluation phase of the proposed systems.

1.1.2 Subtask 2

Subtask 2 is a modified version of Subtask 1, where a ranking of the given set of terms should be produced, based on the degree that the distribution of senses has shifted between the two documents, or time periods. This is referred as Graded Lexical Semantic Change in the literature (Schlechtweg et al., 2020). The difference between the two normalized sense distributions, namely Jensen Shannon Divergence (Lin, 1991), is used as the ranking criterion. Both the corpora and the annotations are the same with the ones used in Subtask 1.

1.2 Data description

The data for the two tasks are the same and consists of four languages with two corpora per language. The four languages are English, German, Latin and Swedish. The two corpora are divided into two different time periods.

Language	Corpora	Period 1	Period 2
English	CCOHA (Alatrash et al., 2020)	1810-1860	1960-2010
German	DTA, BZ and ND [1]	1800-1899	1946-1990
Latin	LatinISE (McGillivray and Kilgarriff, 2013)	-200-0	0-2000
Swedish	Kubhist (Språkbanken,)	1790-1830	1895-1903

Table 1: Data used in the different tasks.

The data has been lemmatized and converted to lowercase. For each of the corpora there are some particularities noted in the SemEval-2020 Task 1 data description[2]. The most important particularity is the frequent OCR errors found in several of the corpora, lowering the quality of the data.

2 Solution

2.1 Solution Outline & Main Idea

Given a word W we generate the contextualized embeddings for all occurrences of W in the two corpora (C_1 and C_2) while keeping a reference to the source corpora. The contextualized embeddings from both corpora are then clustered. Each occurrence of a word is thus represented by its contextualized embedding, source label and cluster label. We then solve the tasks using cluster labels as a direct proxy for senses. We refer to this method as *SenseCluster*.

[1] Berliner Zeitung. Diachronic newspaper corpus published by Staatsbibliothek zu Berlin [online]. 2018. Deutsches Textarchiv. Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache. Herausgegeben von der Berlin-Brandenburgischen Akademie der Wissenschaften [online]. 2017. Neues Deutschland. Diachronic newspaper corpus published by Staatsbibliothek zu Berlin [online]. 2018.

[2] `https://competitions.codalab.org/competitions/20948\#learn\_the\_details-data`

The main idea is that contextualization, in part, serves to disambiguate between senses: we hypothesize that the contextualized embeddings of *cell* in the phone-sense, in general, are closer to each other in embedding space than the contextualized embeddings of *cell* in the chamber-sense, and vice versa. In other words: we hypothesize that the senses of a word W manifests themselves as clusters in the contextualized embeddings of W. The origins of this idea can be traced back to the work of (Schütze, 1998). Recent work in Lexical Semantic Change using BERT gives credence to our hypothesis : (Giulianelli et al., 2020; Martinc et al., 2020) perform k-means clustering on BERT representations of target words in order to detect temporal semantic change in a large diachronic English corpus (Davies, 2012). Additionally, (Wiedemann et al., 2019) show that a simple k-Nearest Neighbor classifier (Cover and Hart, 1967) on contextualized representations can be used for word sense disambiguation.

2.2 Contextualized Embeddings: XLM-R

Word embeddings can handle synonymy, but not polysemy (at least not in any obvious way; but there are some attempts at uncovering polysemy in word embeddings, such as Relative Neighborhood Graphs (Cuba Gyllensten and Sahlgren, 2015)). Contextualized language models on the other hand do; for each occurrence of a term, a contextualized language model will produce a *contextualized* embedding, which takes into account the surrounding context. Prominent examples of contextualized language models are BERT (Devlin et al., 2018) and GPT (Radford et al., 2019). BERT is a Transformer-based model (Vaswani et al., 2017), which produces deep bidirectional representations, as a result of the masked language training objective. On the other hand, GPT is a unidirectional, Transformer-based model (Vaswani et al., 2017), which is pre-trained using the standard language modeling objective (Radford et al., 2019). Pre-trained contextualized language models are often transferred to task-specific architectures (Devlin et al., 2018), (Radford et al., 2019), based on previous work (Howard and Ruder, 2018).

Contextualized language models are extensively used in the domain of cross-lingual language understanding (Lample and Conneau, 2019), (Conneau et al., 2019). Apart from being beneficial for cross-lingual understanding tasks, contextualized cross-lingual embeddings enable model transfer between languages (Ruder et al., 2019). The latter can be beneficial for low-resource languages.

We use XLM-R (Conneau et al., 2019) for producing term representations. XLM-R is a Transformer-based masked language model, trained on 2.5T of filtered CommonCrawl data in 100 languages. Comparing to previous multilingual masked language models, such as multilingual BERT (mBERT) (Devlin et al., 2018) and XLM (Lample and Conneau, 2019), the size of the pre-training dataset of XLM-R is increased by several orders of magnitude, especially for low-resource languages (Conneau et al., 2019). XLM-R outperforms mBERT (Devlin et al., 2018) and XLM (Lample and Conneau, 2019) in cross-lingual classification, as well as monolingual tasks (Conneau et al., 2019).

ISO code	**Language**	**Tokens** (M)	**Size** (GiB)	**ISO code**	**Language**	**Tokens** (M)	**Size** (GiB)
en	English	55608	300.8	**de**	German	10297	66.6
la	Latin	390	2.5	**sv**	Swedish	77.8	12.1

Table 2: Languages and statistics for CC-100 corpora used by XLM-R (Conneau et al., 2019).

We hypothesize that our multilingual setting can benefit from the use of XLM-R, by virtue of cross-lingual transfer. This can be especially advantageous for less resourced languages such as Latin, and perhaps also Swedish.

2.3 Clustering: K-Means

Our approach to the problem of semantic shift detection is cluster-based. We use K-Means++ (Arthur and Vassilvitskii, 2006) to induce the optimal set of sense-clusters in the contextualized embedding space. K-Means++ is a modified version of the widely used K-Means clustering algorithm, which splits the input data points into a predefined set of clusters, by minimizing the in-cluster average square distance (Lloyd, 1982). Trying to alleviate the dependency of K-Means performance on proper initialization of the cluster centroids, K-Means++ introduces a randomized seeding technique (Arthur and Vassilvitskii, 2006).

Previous work (Wiedemann et al., 2019) shows that distance-based methods, such us k-Nearest Neighbor classifier (Cover and Hart, 1967), can be used to group contextualized embeddings which encode the same sense-information. Given the unsupervised nature of our task, K-Means++ is a reasonable choice for our system.

2.4 Method

We generate contextualized embeddings of target words using XLM-R.[3] Given, for example, target word *edge*, and the sentence "they sit down together upon the edge of the bed", the whole sentence is passed as input to XLM-R, we then extract the embedding from the output layer corresponding to the word *edge*, i.e. its *contextualized embedding*. In the case when the target word consists of several wordpieces, and thus several embeddings, we take the average of these. We then cluster all contextualized embeddings for a target term using K-Means++ [4] with the distance metric set to euclidean. For simplicity, we set the number of clusters to 8 for all target terms and languages.

To measure diachronic shift between the two corpora we aggregate this clustering into a table as seen in Table 3 by counting the number of occurrences per cluster label and source.

Word#	Corpus 1		Corpus 2	
Cluster / Sense 1	12	40%	1	3%
Cluster / Sense 2	18	60%	11	37%
Cluster / Sense 3	0	0%	18	60%

Table 3: Example of a cluster assignment for the contextualized embeddings of a word. For Subtask 1 we say that there has been a sense change if there exists a cluster such that it contains < 2 occurrences from corpus 1 and > 5 occurrences from corpus 2, or vice versa. In this example, going from Corpus 1 to Corpus 2, the word lost Sense 1, but gained Sense 3. For Subtask 2 we measure the Jensen Shannon Divergence between the sense distributions of the corpora. In this example, Corpus 1 has sense distribution $(0.4, 0.6, 0)$, whereas Corpus 2 has sense distribution $(0.03, 0.37, 0.6)$, which results in a Jensen Shannon Divergence of ≈ 0.73.

2.5 Subtask 1

Using the cluster labels as a proxy for senses, we solve the first task using the method described in the task reference (Schlechtweg et al., 2020). If there exists a cluster such that it contains $< k$ occurrences from corpus 1 and $> n$ occurrences form corpus 2, or vice versa, we say that there has been a sense change.

We always let $k = 2$, $n = 5$, and set the number of cluster to 8, regardless of language and the total number of occurrences. For example, given the cluster assignments in table 3 we would say that there has been two sense changes: Going from Corpus 1 to Corpus 2, the word lost Sense 1, but gained Sense 3.

We consider this our baseline approach and while the hyperparameters for the number of clusters, k and n can be tuned, we believe that a different choice of clustering algorithm would yield larger improvements in performance.

2.6 Subtask 2

Subtask 2 is also solved by a direct translation of the task definition (Schlechtweg et al., 2020), i.e. we solve it by computing the Jensen Shannon Divergence between the cluster distributions of the two corpora. We use the same cluster assignments in Subtask 2 as in Subtask 1.

3 Results

The code for the experiments is made publicly available[5]. Table 4 and 5 show the results of the top three submissions, our submission, and the best performing baseline on Subtask 1 and Subtask 2. In both cases,

[3] https://github.com/pytorch/fairseq/tree/master/examples/xlmr

[4] https://scikit-learn.org/stable/modules/clustering.html#k-means

[5] https://github.com/Apsod/sensecluster

	Score				
Team	All	English	German	Latin	Swedish
UWB	0.687 (1)	0.622	0.750	0.700	0.677
Life-Language	0.686 (2)	0.703	0.750	0.550	0.742
Jiaxin & Jinan	0.665 (3)	0.649	0.729	0.700	0.581
Skurt	0.629 (9)	0.568	0.562	0.675	0.710
Baseline (CNT+CI+CD)	0.613 (11)	0.595	0.688	0.525	0.645

Table 4: Subtask 1 results (Accuracy). Our method was the ninth most performant method in the evaluation phase.

	Score				
Team	All	English	German	Latin	Swedish
UG Student Intern	0.527 (1)	0.422	0.725	0.412	0.547
Jiaxin & Jinan	0.517 (2)	0.325	0.717	0.440	0.588
cs2020	0.503 (3)	0.375	0.702	0.399	0.536
Skurt	0.374 (7)	0.209	0.656	0.399	0.234
Baseline (CNT+CI+CD)	0.144 (18)	0.022	0.216	0.359	-0.022

Table 5: Subtask 2 results (Spearman rank correlation). Our method was the seventh most performant method in the evaluation phase.

the best performing baseline is the `CNT+CI+CD` model, a co-occurrence counting method (Schlechtweg et al., 2020). For Subtask 1 our method outperforms the best performing baseline on average, but performs worse for English and German. For Subtask 2 our method outperforms the best performing baseline for all languages by a wide margin except for Latin, where the performance is only slightly better.

4 Discussion & Conclusion

We chose XLM-R because it is a pre-trained, performant, single, contextualized model trained on all the languages in the task. As such, the method easily extends to all other languages XLM-R has been trained on. However, the choice of XLM-R has certain drawbacks. It is trained on CommonCrawl data, which we assume is heavily skewed towards contemporary language. This might have a negative impact on model performance, since the task is dominated by historical data. More importantly, we believe that the biggest drawback of using XLM-R (or similar language models) is that it is trained with minimal preprocessing. The task data, on the other hand, was heavily lemmatized, PoS-tagged (for English), and had frequent OCR errors. We believe this mismatch in preprocessing methods and data quality has had a very detrimental effect on the quality of the contextualized embeddings we extract from XLM-R.

Our approach, while not yielding great results, outperformed the baselines and scored relatively well on task 2. We argue that it performed surprisingly well given the simplicity of the approach: our method can be condensed into:

(i) the hypothesis that clusters of tokens in contextualized embedding space approximate senses (a conjecture that is also further corroborated by other recent work (Wiedemann et al., 2019)),

(ii) the implicit (and completely preposterous) assumption that every term has eight senses

If we assume that (i) is true, then the apparent falsehood of (ii) poses two problems: if a term has more than eight senses, our method will conflate senses by putting them in the same cluster, a *supersense*, if you will. If it has less than eight senses, our method will split senses into *subsenses*. Both of these scenarios are problematic for our model: a subsense change might occur without a sense change, and conversely, a sense change might occur within a supersense without a supersense change. We hypothesize that this effect is greater in Subtask 1 than in Subtask 2, due to the more discrete nature of Subtask 1. One possible remedy to this is to use a data driven method to determine the number of clusters, for example X-means

(Pelleg et al., 2000) or Affinity Propagation (Frey and Dueck, 2007), rather than choosing an arbitrary constant. An alternative direction could be to employ non-parametric density-based clustering methods, such as DBSCAN (Ester et al., 1996).

Based on the simplicity and shortcomings of the clustering approach we believe the relatively good performance can be attributed to the use of contextualized language models, and by extension that (i) is at least partly true. We believe that by improving the second step, i.e. the grouping of sets of contextualized embeddings into appropriate clusters, we could improve performance significantly.

References

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2020. Ccoha: Clean corpus of historical american english. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6958–6966.

David Arthur and Sergei Vassilvitskii. 2006. k-means++: The advantages of careful seeding. Technical report, Stanford.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Amaru Cuba Gyllensten and Magnus Sahlgren. 2015. Navigating the semantic horizon using relative neighborhood graphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2451–2460, Lisbon, Portugal, September. Association for Computational Linguistics.

Mark Davies. 2012. Expanding horizons in historical linguistics with the 400-million word corpus of historical american english. *Corpora*, 7(2):121–157.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. *arXiv preprint arXiv:2004.14118*.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany, August. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. *arXiv preprint arXiv:1806.03537*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020. Capturing evolution in word usage: Just add more clusters? In *Companion Proceedings of the Web Conference 2020*, pages 343–349.

Barbara McGillivray and Adam Kilgarriff. 2013. Tools for historical corpus research, and a corpus of latin. *New Methods in Historical Corpus Linguistics*, (3):247–257.

Dan Pelleg, Andrew W Moore, et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Guy D Rosin, Eytan Adar, and Kira Radinsky. 2017. Learning word relatedness over time. *arXiv preprint arXiv:1707.08081*.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Eyal Sagi, Stefan Kaufmann, and Brady Clark. 2008. Tracing semantic change with latent semantic analysis. In *In Proceedings of ICEHL 2008*.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *To appear in Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Språkbanken. The Kubhist Corpus v2. Språkbanken Text, Department of Swedish, University of Gothenburg, filter by "Kubhist2 . Online; Downloaded in 2019.

Terrence Szymanski. 2017. Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: short papers)*, pages 448–453.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to lexical semantic change. *arXiv preprint arXiv:1811.06278*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 673–681, New York, NY, USA. Association for Computing Machinery.